

AMENDMENTS TO THE CLAIMS

1 1. (currently amended) A method for dynamically allocating data buffers to a data
2 structure, comprising the computer-implemented steps of:
3 assigning a logging thread to said data structure, wherein said logging thread is
4 configured to insert free data buffers into said data structure;
5 monitoring an amount of log data that is being stored within data buffers
6 associated with said data structure, wherein the step of monitoring the
7 amount of log data comprises the step of said logging thread tracking how
8 often said data buffers associated with said data structure are determined
9 to be full;
10 based on the amount of log data that is being stored within said data buffers,
11 determining whether additional data buffers need to be linked into said
12 data structure; and
13 if additional data buffers need to be linked to said data structure,
14 identifying one or more free buffers; and
15 linking said one or more free data buffers into said data structure.

1 2. (original) The method of Claim 1, further comprising the steps of:
2 receiving requests for content that is associated with a web site domain;
3 generating log data based on the requests; and
4 writing said log data in one or more data buffers associated with said data
5 structure.

1 3. (original) The method of Claim 2, wherein the step of monitoring an amount of
2 data that is being stored within data buffers includes the step of tracking how
3 many requests are being received for content that is associated with said web site
4 domain.

1 4. (original) The method of Claim 1, further comprising the steps of:

2 determining that a particular data buffer should be removed from said data
3 structure;
4 unlinking said particular data buffer from said data structure; and
5 inserting said particular data buffer into a ready-to-write buffer list.

1 5. (original) The method of Claim 4, wherein the step of determining that a
2 particular data buffer should be removed comprises the step of detecting that said
3 particular data buffer is full.

1 6. (original) The method of Claim 4, wherein the step of determining that a
2 particular data buffer should be removed comprises the step of detecting that said
3 particular data buffer has not been removed from said data structure for a
4 particular period of time.

1 7. (original) The method of Claim 4, further comprising the steps of:
2 removing said particular data buffer from said ready-to-write buffer list, wherein
3 said ready-to-write buffer list is located within a first memory area;
4 storing log data information in said particular data buffer to a second memory
5 area, wherein said second memory area is distinct from said first memory
6 area; and

7 inserting said particular data buffer into a free buffer pool, wherein said free
8 buffer pool maintains free data buffers that may be inserted into any one
9 of a plurality of data structures that are each associated with a particular
10 web site domain.

1 8. (original) The method of Claim 7, wherein:
2 the step of identifying one or more free buffers comprises the step of selecting one
3 or more free buffers from said free buffer pool; and
4 the step of linking said one or more free data buffers into said data structure
5 comprises the steps of,
6 identifying one or more entries in said data structure; and

7 linking said one or more free data buffers into said one or more entries in
8 said data structure.

1 9. (original) The method of Claim 4, wherein:
2 said log data is generated based on request that are received for content associated
3 with a particular web site domain; and
4 said step of inserting said particular data buffer comprises the step of linking said
5 particular data buffer into a queue that maintains only data buffers that
6 contain log data associated with requests for said particular web site
7 domain.

1 10. (canceled)

1 11. (currently amended) A method for dynamically allocating data buffers in a web
2 server, comprising the computer-implemented steps of:
3 configuring said web server to service requests for multiple web site domains;
4 assigning a buffer management structure to a particular web site domain of said
5 multiple web site domains, wherein said buffer management structure
6 includes a data structure that maintains links to data buffers used to buffer
7 log data that is based on content requests that are directed to said
8 particular web site domain; and
9 dynamically controlling how many data buffers are allocated to said data structure
10 based on how many content requests said web server receives for said
11 particular web site domain;
12 wherein the step of dynamically controlling how many data buffers are allocated
13 to said data structure comprises the steps of:
14 monitoring how often data buffers are inserted into a ready-to-write buffer
15 list; and

16 dynamically adjusting an amount of data buffers that are associated with
17 said data structure based on how often data buffers are inserted
18 into said ready-to-write buffer list.

1 12. (currently amended) The method of Claim 11, further comprising the steps of:
2 in response to determining that a particular data buffer should be removed from
3 said data structure,
4 removing said particular data buffer from said data structure; and
5 inserting said particular data buffer into [[a]]said ready-to-write buffer list;
6 and wherein the step of dynamically controlling how many data buffers are
7 allocated to said data structure comprises the steps of:
8 monitoring how often data buffers that are inserted into said ready to write
9 buffer list; and
10 dynamically adjusting an amount of data buffers that are associated with
11 said data structure based on how often data buffers are inserted
12 into said ready to write buffer list.

1 13. (currently amended) A computer-readable medium carrying one or more
2 sequences of instructions for dynamically allocating data buffers to a data
3 structure, wherein execution of the one or more sequences of instructions by one
4 or more processors causes the one or more processors to perform the steps of:
5 assigning a logging thread to said data structure, wherein said logging thread is
6 configured to insert free data buffers into said data structure;
7 monitoring an amount of log data that is being stored within data buffers
8 associated with said data structure, wherein the step of monitoring the
9 amount of log data comprises the step of said logging thread tracking how
10 often said data buffers associated with said data structure are determined
11 to be full;

12 based on the amount of log data that is being stored within said data buffers,
13 determining whether additional data buffers need to be linked into said
14 data structure; and
15 if additional data buffers need to be linked to said data structure,
16 identifying one or more free buffers; and
17 linking said one or more free data buffers into said data structure.

1 14. (original) The computer-readable medium of Claim 13, further comprising
2 instructions for performing the steps of:
3 receiving requests for content that is associated with a web site domain;
4 generating log data based on the requests; and
5 writing said log data in one or more data buffers associated with said data
6 structure.

1 15. (original) The computer-readable medium of Claim 14, wherein the step of
2 monitoring an amount of data that is being stored within data buffers includes the
3 step of tracking how many requests are being received for content that is
4 associated with said web site domain.

1 16. (original) The computer-readable medium of Claim 13, further comprising
2 instructions for performing the steps of:
3 determining that a particular data buffer should be removed from said data
4 structure;
5 unlinking said particular data buffer from said data structure; and
6 inserting said particular data buffer into a ready-to-write buffer list.

1 17. (original) The computer-readable medium of Claim 16, wherein the step of
2 determining that a particular data buffer should be removed comprises the step of
3 detecting that said particular data buffer is full.

1 18. (original) The computer-readable medium of Claim 16, wherein the step of
2 determining that a particular data buffer should be removed comprises the step of

3 detecting that said particular data buffer has not been removed from said data
4 structure for a particular period of time.

1 19. (original) The computer-readable medium of Claim 16, further comprising
2 instructions for performing the steps of:
3 removing said particular data buffer from said ready-to-write buffer list, wherein
4 said ready-to-write buffer list is located within a first memory area;
5 storing log data information in said particular data buffer to a second memory
6 area, wherein said second memory area is distinct from said first memory
7 area; and
8 inserting said particular data buffer into a free buffer pool, wherein said free
9 buffer pool maintains free data buffers that may be inserted into any one
10 of a plurality of data structures that are each associated with a particular
11 web site domain.

1 20. (original) The computer-readable medium of Claim 19, wherein:
2 the step of identifying one or more free buffers comprises the step of selecting one
3 or more free buffers from said free buffer pool; and
4 the step of linking said one or more free data buffers into said data structure
5 comprises the steps of,
6 identifying one or more entries in said data structure; and
7 linking said one or more free data buffers into said one or more entries in
8 said data structure.

1 21. (original) The computer-readable medium of Claim 16, wherein:
2 said log data is generated based on request that are received for content associated
3 with a particular web site domain; and
4 said step of inserting said particular data buffer comprises the step of linking said
5 particular data buffer into a queue that maintains only data buffers that

6 contain log data associated with requests for said particular web site
7 domain.

1 22. (canceled)

2

1 23. (currently amended) A computer-readable medium carrying one or more
2 sequences of instructions for dynamically allocating data buffers in a web server,
3 wherein execution of the one or more sequences of instructions by one or more
4 processors causes the one or more processors to perform the steps of:
5 configuring said web server to service requests for multiple web site domains;
6 assigning a buffer management structure to a particular web site domain of said
7 multiple web site domains, wherein said buffer management structure
8 includes a data structure that maintains links to data buffers used to buffer
9 log data that is based on content requests that are directed to said
10 particular web site domain; and
11 dynamically controlling how many data buffers are allocated to said data structure
12 based on how many content requests said web server receives for said
13 particular web site domain;

14 wherein the step of dynamically controlling how many data buffers are allocated
15 to said data structure comprises the steps of:

16 monitoring how often data buffers are inserted into a ready-to-write buffer
17 list; and

18 dynamically adjusting an amount of data buffers that are associated with
19 said data structure based on how often data buffers are inserted
20 into said ready-to-write buffer list.

1 24. (currently amended) The computer-readable medium of Claim 23, further comprising
2 instructions for performing the steps of:

3 in response to determining that a particular data buffer should be removed from said
4 data structure,
5 removing said particular data buffer from said data structure; and
6 inserting said particular data buffer into [[a]]said ready-to-write buffer list;
7 and ~~wherein the step of dynamically controlling how many data buffers are allocated~~
8 ~~to said data structure comprises the steps of:~~
9 ~~monitoring how often data buffers that are inserted into said ready to write~~
10 ~~buffer list; and~~
11 ~~dynamically adjusting an amount of data buffers that are associated with said~~
12 ~~data structure based on how often data buffers are inserted into said~~
13 ~~ready to write buffer list.~~
14